Journal of the Korean Institute of Illuminating and Electrical Installation Engineers Vol. 21, No.9, pp. 81~86 November 2007 (21-9-12)

Distribution Network Switching Automation Using Active Web Based Management

Sang-Yule Choi*

Abstract

Electric utility companies have the responsibility of providing good electricity for their customers. They have introduced the DAS (Distribution Automation System) to automate the power distribution networks. DAS engineers require state-of-the-art applications, such as a way to actively manage the distribution system and gain economic benefits from a flexible DAS architectural design. The existing DAS is not capable of handling these needs. It requires operator intervention whenever feeder overloading is detected while operator error could cause the feeder overload area to be extended. It also utilizes a closed architecture and it is therefore difficult to meet the system migration and future enhancement requirements. This paper represents a web based, platform-independent, flexible DAS architectural design and active database application. Recent advanced Internet technologies are fully utilized in this new DAS architecture allowing it to meet the system migration and future enhancement requirements. By using an active database, the DAS can minimize the feeder overloading area in the distribution system without operator intervention, thereby minimizing mistakes due to operator error.

Key Words : Power IT, DAS, Active Management, Platform-independent

1. Introduction

Electrical utility companies have the responsibility of providing quality electricity for their customers. Therefore, the DAS (Distribution Automation System) has been introduced to control and operate the complex power distribution system in an economically and reliably. It includes important functions such as feeder automation, load control and telemetering. In Korea, the electrical utility company is now facing deregu-lation and privatization. Privatized distribution companies will appear in the next few years. These companies will need a new way of thinking and new solutions for the open-access, competitive electricity market. The existing DAS is unable to meet these new requirements because it has to be managed by passively and it utilizes closed system architecture.

In closed system architecture, DAS utilizes patented software which is tightly coupled to a particular operating environment. Therefore,

* Main author : Department of Information and Mechartronics Science, Induk University.
Tel : +82-2-950-7427, Fax : +82-2-950-7439
E-mail : ppk99@induk.ac.kr
Date of submit : 2007. 9. 13
First assessment : 2007. 9. 18
Completion of assessment : 2007. 10. 8

system integration and data migration in a heterogeneous environment pressures the DAS developers and operators. Due to passive DAS management, DAS is always exposed to unintentional operator error. Therefore, whenever a feeder overloading is detected, an inexperienced operator may cause the situation to become worse.

Despite these defects, the closed architecture and passive management have been successfully applied to DAS. However, under an open-access, competitive market environment, data migration in a heterogeneous environment and reliable active system management becomes critical issues for distribution utility companies. This environment causes a need on the part of DAS engineers for a set of state-of-the-art applications, such as using an Internet application for convenient data exchange and system openness and adopting an active database application for reliable, active system management. Based on these requirements, a new DAS architecture based on an open system and an active rule application is presented. Open DAS system architecture utilizes recently-developed Internet technology and an active relational database.

2. Definition of Rules for a DAS Database

The data acquisition system which is located in distribution networks detects currents, section loadings, voltage and the status of close/open in real time, and the present attribute values of the switch is updated by detected ones. An active DAS database monitors the present state of distribution networks by using updated values: If an updated value triggers an event and the conditions are true, then this implies that the distribution networks are overloaded. Therefore action is taken for feeder reconfiguration to relieve the feeder overload and minimize line loss. Switches to be opened (or closed) are selected after feeder reconfiguration is executed.

Feeder reconfiguration can fire other rules which update the attributed close/open status value of the selected switches and send close/open signals to appropriate intelligent switches installed in distribution networks.

Due to maintaining a radial distribution networks structure, the updated value of the switch can trigger other rules that update the close/open status value of another switch (not a one selected from feeder reconfiguration). The updated attribute value of the switch can trigger rules which update the close/open status value of the appropriate feeder section connected to the updated switch since the close/open status of the feeder section is dependent on the attributed values of the appropriate switch object. Distribution networks can be operated with minimum losses, and a radial structure can be maintained without operator intervention thanks to the definition of the active rules which are able to disseminate updates.

The definitions of the active rules for DAS are as follows:

Rule 1) If the initially attributed close/open status value of the switch is changed from open to close then the attributed close/open status value of the appropriate feeder section connected to the switch is changed from open to close

Rule R1 for switch

Event: update to st

Condition: updated(switch(S)), NEW S.st=close

Action: update feeder section.st=close where feeder section.ssn = switch.ssn

Rule 2) If the initially attributed close/open status value of a switch object is changed from close to open then the close/open status value of the appropriate feeder section object connected to the switch is changed from close to open.

Rule R2 for switch

Event: update to st

Condition: updated(switch(S)), NEW S.st=open

Action: update feeder section.st=open where feeder section.ssn = switch.ssn

Rule 3) If the attributed loadings value of a switch object is updated then the section loadings value of the feeder section is updated by the same value.

Rule R3 for switch

Event: update to ssnmva

Condition: True

Action: update feeder section.ssnmva = switch. ssnmva where feeder section.ssn = switch.ssn

Rule 4) If the updated section loadings value of a feeder section is exceeded by 80[%] of the feeder capacity then the feeder reconfiguration program is executed to minimize line losses and relieve overload.

Rule R4 for feeder section

Event: update to ssnmva

Condition: updated(feeder section(FD)),

NEW FD.ssnmva > FD.fnc

Action: reconfiguration()

Rule 5) If reconfiguration() is executed and the switch to be closed (or opened) is selected, then the attributed close/open status of selected switch is updated as closed (or open).

Rule R5 for reconfiguration()

Event: reconfiguration()

Conditon: TRUE

Action: (update switch.st=open where switch.ssn

= result of open reconfiguration())

&&(update switch.st=close where switch.ssn= result of close reconfiguration())

where, the result of open reconfiguration(): the selected switch to be opened resulting from feeder reconfiguration.

result of close reconfiguration(): the selected switch to be closed resulting from feeder reconfiguration.

Rule 6) If the attributed voltage value of a switch is updated then the attributed voltage value of the appropriate feeder section is updated

Rule R6 for switch

Event: update vv

Condition: updated(switch(S)), TRUE

Action: update feeder section.vv =

switch.vv—Vdrop

where, feeder section.ssn = switch.ssn

Where, Vdrop : Voltage drop of feeder section

Rule 7) If the updated voltage value of a feeder section is lower than the minimum voltage value of the feeder section then the feeder reconfiguration program is executed to minimize line losses and relieve overload.

Rule R7 for feeder section

Event: update to vv

Condition: updated(feeder section(FD)),

NEW FD.vv > FD.min_vv

Action: reconfiguration()

Rule 8) If the initially attributed close/open status value of a switch is changed from open to close then the active database sends a signal to the intelligent switch to be closed

Rule R8 for switch

Event: update to st

Condition: updated(switch(S)), NEW S.st=close

Action: signals intelligent switch to be closed

Rule 9) If the initially attributed close/open status value of a switch is changed from close to open, the active database sends a signal to the appropriate intelligent switch to be opened

Rule R9 for switch

Event: update to st

Condition: updated(switch(S)), NEW S.st=open

Action: signal to intelligent switch to be opened

Rule 10) If feeder reconfiguration is executed then the information is updated with the resultant data from the feeder reconfiguration.

Rule R10 for reconfiguration

Event: reconfiguration()

Condition: TRUE

Action:v(update information.rsn = history) && (update information.ploss= result loss of reconfiguraion())&&(update information.dploss = result of change_of_loss of reconfiguration())

Where, history = reconfiguration times

result loss of reconfiguraion(): total loss after feeder reconfiguration

result of change_of_loss of reconfiguration(): the amount of loss change resulting from feeder reconfiguration;

3. Active Web based DAS database system architecture

The requirements that a flexible open DAS system architecture must satisfy are two-fold [3]. It is built to vendor-neutral standards (ease of software use).

It must provide the ability to enhance the existing DAS without relying on one vendor (ease of continuing development and maintenance of the software package)

These requirements can be met by using the Internet as the operating environment.

Using Internet technology for DAS architecture will allow the following benefits:

First, it supports cross-platform architecture:

In a standardized Internet browser environment with HTML and TCP/IP protocols, users will continue using the platforms with which they are most familiar without being conscious of a different hardware platform.

Second, it follows open system standards:

By following Structured Query Language (SQL), HTML, HTTP, FTP, TCP/IP, and PPP, data exchange and system expansion are easily achieved with minimum effort[4].

The proposed new web based DAS architecture make use of the Java 2 Enterprise Edition architecture which is a Web-based multitier architecture, as presented in Fig. 1.

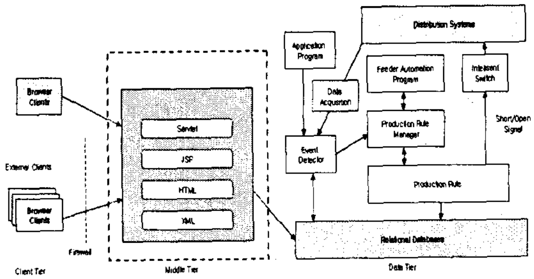

Fig. 1. A system architecture for active web based DAS database system

This architecture can be expressed by subdividing the two sections of Fig. 1.

One section is web based architecture which supports an open system and the other is an active database architecture which supports DAS feeder automation.

For web based architecture, the structure can be divided into three tiers[3].

Client tier: This provides user interface

Middle tier: This is subdivided into the Web server and the application server.

Data tier: This handles information storage

In the middle tier, the application server transfers a request from the web into the appropriate functions in the system and provides for the interface of different kinds of databases. The Web server acts as the gateway for Web-based clients to access the database.

Power distribution engineers and operators send HTTP requests to the Web server through an industrial standard web browser. If a web page containing a database SQL command is requested, the database interprets the SQL commands and returns matching data from the database back to the Web server. The matching data are formulated as a web page, which is displayed in the client window[4].

For active database architecture shown on the right-side of Fig. 1:

The structure can be expressed specifically by being divided into three parts:

- Event detector: This checks the telemetered data and sends signals to the production rule manager if integrity constraints are violated.
- Production rule manager

This accepts signals from the event detector and determines which rules are to be fired.

- Production rule

This allows the specification of data manipulation operations that are executed automatically whenever certain events occur or certain conditions are satisfied.

The data acquisition system located in the distribution network detects the close/open status of a switch, current, voltage and load, and sends the detected data to the central DAS database system.

In a DAS database system, the event detector accepts the detected data sent from the data acquisition system and sends data modification operations to the database. If integrity constraints are violated because of data modification then the event detector sends a signal to the production rule manager. By using the production rule manger, production rules are fired and the resultant data are updated in the database. After updating values in the database, the event detector verifies if the updated values satisfy integrity constraints or not. If integrity constraints are violated then the event detector sends signals to the production rule manager. When Rule R10 and Rule R5 are triggered, close/open signals are sent to the appropriate intelligent switch located in the distribution networks.

4. DEVELOPMENT OF a WEB BASED POWER DISTRIBUTION SYSTEM

4.1 Initial Man-Machine Interface

In this paper, a web based power distribution system is implemented using the JAVA, Html, and Oracle DBMS.

The Target distribution system is KPECO's Seoul K section 180 distribution network. The initial Man-Machine Interface is as follows in Fig 2.

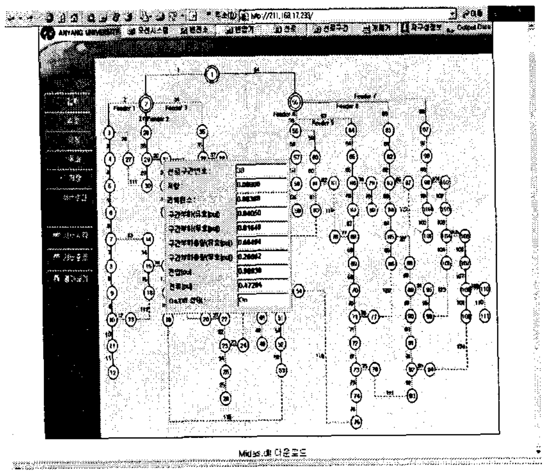

Fig. 2. Example of initial windows application program

4.2 Power Distribution Automation Using an Active Web Based Database

If Feeder 1 is overloaded to 17[MVA}, the intelligent switch with which the Feeder is equipped detects the overload and sends the data to the central database, which updates the attributed load of the Feeder. if the overload exceeds 80[%] of the overload limit then Rule 15 is triggered to activate the program Cyclic_Reconfiguration()

After Cyclic_Reconfiguration( ) is activated, initial on switches {16, 36, 42, 49, 67, 100} are turned off and Initial off switches {113, 114, 115, 116, 119, 122} are turned on. The changing of switch status triggers Rule 9, Rule 10, Rule 19, and Rule 20.

After the switches are activated, feeder reconfiguration is as follows in Fig 3.

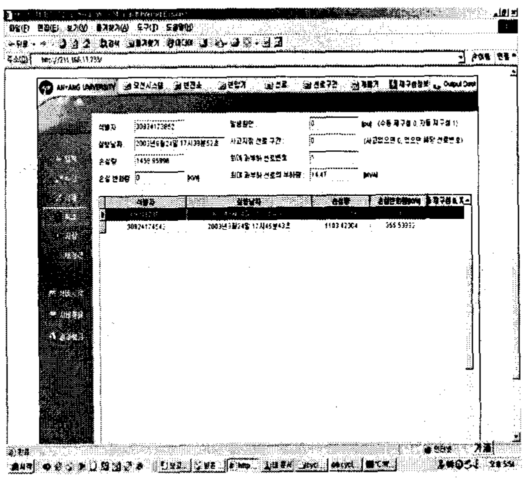

Fig. 3. Reconfiguration information table

5. Conclusion

This paper presents a new active Web-based database architectural design for automatic distribution system applications. This architecture includes web-based architecture and active database sections. For the web-based sections, the Java 2 Enterprise Edition architecture for open systems was utilized for its ease of continuing development and maintenance of the software package. This will make it easier to meet the challenges of a competitive, open-access market environment. For the active database architecture, a production rule and production rule manager for DAS feeder automation was designed by utilizing the proposed rules. The distribution network can be operated reliably with minimum operator intervention.

Acknowledgements

The research was supported by the Induk Institute of Technology

References

[1] J. Widom, S.Ceri: Active Database Systems: Triggers and rules for Advanced Database Processing. MORGAN KAUFMANN PUBLISHERS.

[2] E. Baralis, S. Ceri and S.Paraboshi: Compile-Time and Runtime Analysis of Active Behavior. IEEE Trans, On Knowledge and Data Engineering, Vol. 10, No.3, (1998) 353 - 370.

[3] S.Chen, F.Y. LU : Web-Based Simulations of Power Systems. IEEE Computer Application in Power, January (2002) 35-40.

[4]J.T. Ma, T. Liu, L.F. Wu: New Energy Management System Architectural Design and Intranet/Internet Applications to Power Systems Conference Proc. Power industry computer application conference. (1995) 207- 212.

[5] E.Baralis, A.Bianco: Performance Evaluation of Rule Execution Semantics in Active Database. Tech. Rep. DAI.EB. 96.1, Aug. (1996).

[6] IEEE Task Force on Power System Control Center Database: Critical Issues Affecting Power System Control Center Database. IEEE Trans. On Power System, vol, 11, no.2 , May (1996).

[7] C.S. Martire, D.J.H. Nuttall: Open Systems and Database. IEEE Trans, On Power Sys-tem, Vol. 8, NO. 2, May. (1993).

Biography

Sang-Yule Choi

Born in August 24, 1970. Mr. Choi graduated from the Department of Electrical Engineering, Sungkyunkwan University in 1996. He graduated from the School of Electrical, Electronic and Computer Engineering in the Graduate School of Sungkyunkwan University in 2002 (Ph.D., Engineering). 2002~2004: Full-time lecturer in the School of Digital Media Science, Anyang University. 2004~present: Full-time lecturer in the Department of Information and Mechartronics Science, Induk University.